Serial No. 10/712,463
Examiner's Answer Dated: April 1, 2009
Reply Brief Filed: May 29, 2009

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s):	Judith E. Schwabe et al.		
Assignee:	Sun Microsystems, Inc.		
Title:	OPTIMIZATION OF N-BASE TYPED ARITHMETIC INSTRUCTIONS VIA REWORK		
Serial No.:	10/712,463	Filed:	November 12, 2003
Examiner:	Tuan A. Vu	Group Art Unit:	2193
Docket No.:	P-4181CIP		

Monterey, CA
May 29, 2009

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REPLY BRIEF

Dear Sir:

Pursuant to 37 CFR § 41.41, Appellant files this Reply Brief in response to the Examiner's Answer dated April 1, 2009.

STATUS OF CLAIMS

Claims 1 to 78 are pending. Claims 1 to 78 stand rejected in the Final Office Action of March 27, 2008. The rejections of Claims 1 to 78 have been appealed.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether Claims 1 to 10, 14 to 29, 33 to 48, 52, 53 to 67, 71 to 78 are unpatentable under 35 U.S.C. 103(a) over U.S. Patent No. 5,740,441, hereinafter referred to as Yellin, in view of U.S. Patent No. 6,308,317, hereinafter referred to as Wilkinson?

2. Whether Claims 11 to 13, 30 to 32, 49 to 51, 68 to 70, are unpatentable under 35 U.S.C. 103(a) over U.S. Patent No. 5,740,441, hereinafter referred to as Yellin, in view of U.S. Patent No. 6,308,317, hereinafter referred to as Wilkinson?

ARGUMENT

1. CLAIMS 1 TO 10, 14 TO 29, 33 TO 48, 52 TO 67, 71 TO 78 ARE PATENTABLE

The Examiner's Answer continues the misinterpretations and mischaracterizations of the claims and makes the claim that Appellant's remarks are misdirected. The Office has failed to cite any teaching in combination of reference that renders the matching operation of the claim obvious. As noted at pgs. 20 and 21 of Appellant's brief, the original interpretation of the combination of references was that the references suggested exactly the opposite of what was recited in the claims.

The Examiner's Answer completely redefines the express claim limitations so as to attempt to justify this erroneous rejection. The errors are so numerous that Appellant first addresses the incorrect claim interpretation that is put forth based on Sections (A) to (C) of the Examiner's answer at pages 16 to 21 and second address incorrect characterizations of claim language and the incorrect rationale for ignoring explicit claim limitations.

The Examiner's Answer, in at least Sections (A) to (C), incorrectly goes to some length to assert "The matching limitation should be understood and has been interpreted as subsumed under (or being a complementary action with) the converting step." {Examiner's Answer (C) at pg 20.}

Appellant notes that no 112 rejections are pending in the rejection. Accordingly, the extended comments in this section of the Examiner's Answer concerning the lack of support for and interpretation of the claims are directly contradicted by the absence of such rejections.

Using Claim 1 as an example, Appellant points out that the recited three operations are separate and distinct and are supported at least directly by Fig. 16. Note that in view of an objection to Claim 1 in the paper dated 5/21/2007 at pg. 2 and the statement on pg. 2 of the Advisory Action dated 8/10/2007, the "optimizing" of Fig. 16 was changed to "converting" in the claims. Thus, the converting process in the claim is described as "optimizing" in Fig. 16 and the description thereof. The Office effectively required this change in terminology in the second operation of the claim despite the express teachings in the specification.

Therefore, based on Fig. 16 and the description thereof, there is no basis for pulling the matching operation into the converting operation and to the extent that the Examiner's Answer attempts to combine the two {See for example, Pg. 20 (C), Pg. 23} it is direct evidence that the Claims are not being considered properly and puts the claims into a form that is inconsistent with the specification, which is direct evidence of an improper interpretation.

The MPEP provides "During patent examination, the pending claims must be 'given their broadest reasonable interpretation consistent with the specification.'" MPEP § 2111, 8th Ed. Rev. 6, pg. 2100-37, (Sept. 2007). The Examiner's Answer itself demonstrates that the interpretation used is not supported by the specification and so is not the broadest reasonable interpretation.

Moreover, the interpretation in the Examiner's Answer ignores the plain meaning of the claims as the claims would be interpreted by those of skill in the art in view of the specification. "The broadest reasonable interpretation of the claims must also be consistent with the interpretation that those skilled in the art would reach." Id. at pg. 2100-38.

As recited in the Claim, the converting operation converts a first instruction to a second instruction that operates on an operand of a second type. This converting is done, as recited in the claim, without consideration of the operand that is actually supplied for the second instruction.

The instruction that originally provided the operand that now is processed by the second instruction may provide an operand of a type that is smaller than the second type. Accordingly, the matching operation recites how the process determines whether this situation occurs and if the situation occurs, how the situation is handled. Moreover, the antecedent basis in the matching operation establishes that the matching operation occurs after the converting operation, because the matching operation uses information from the converting operation.

Following the conversion, the matching matches the second type, which is the operand type used by the second instruction, with an operand type of at least one input stack associated with the second instruction. If the operand type is less than the second type, it means that the instruction type for the instruction providing that operand needs to be changed so that an operand of the second type is supplied by that instruction. Accordingly, the claim specifically recites that all instructions between the instruction that is the source of the operand and the second instruction are changed to instructions of the second type, which corrects the situation when created in the converting operation.

Paragraph [0068] at pages 36 and 37 of the Specification expressly describes this at lines 5 and 6 on page 36. Paragraph [0068] states that the paragraph describes operation 2025 of Fig. 20. Paragraph [0067] at page 36 describes that Fig. 20 provides more detail for operation 1630 in Fig. 16, which is the matching operation.

The Examiner's Answer completely mischaracterizes the plain meaning of the claims in attempt to justify ignoring the explicit claim limitation that the type of instruction in a chain of instructions is changed to the second type if the recited operand type is less than the second type.

Thus, the Examiner's Answer not only failed to interpret the claims in view of the level of skill in the art, but also recasts the claims in a way that has nothing to do with the claim language itself or the description in the specification.

At page 17, the Examiner's Answer stated:

> One of ordinary skill in art cannot see how a full instruction type which operates on operands can be made **equal to** an operand type.

The Examiner's answer goes on to completely contradict knowledge that is well known in the art with respect to operand types, instruction types, etc. as established in the instant application. Even if such knowledge concerning operand types and instruction types were not well known, numerous specific examples are described in the detailed description on such types how to make such comparisons of such types.

For example, Paragraph [0052], lines 5 to 10 at pg. 28 of the description provides:

> . . . . If the inputs to the instruction are different sized types, the smaller-typed inputs are changed to equal the larger-typed inputs. By way of example, if one instruction input is type "int" and another instruction input is type "short", the type "short" instruction input is changed to an "int" type. This process continues recursively until the origin of the smaller type and all of its subsequent instructions are changed to the larger type

As is known, the inputs to the instruction referred to in this section are operands. The two types are "int" and "short" The description states that the smaller type is "short." The process goes through a chain that starts with the instruction that is the origin of the operand with the smaller type "short" and all the subsequent instructions are changed to the larger type "int." This demonstrates how to compare an operand type to an instruction type and how to change the type of instructions in a chain based on the comparison. This alone is sufficient to demonstrate that the above quoted statement in the Examiner's Answer is without merit, because it specifically describes how a less than test is performed between an operand type and another type and how instruction types are changes based on the comparison.

In addition, instruction types and operand types for arithmetic expressions are terms well known to those of skill in the art. For example, in the JAVA programming language, the instruction type is represented explicitly in the opcode mnemonic by a letter: i for an int operation; l for long; s for short; b for byte; c for char; f for float; d for double; and a for reference. Paragraphs [004] to [018] of the Specification include a discussion of these types as well as instruction types and operand types in other computer programming languages.

For the JAVA programming language example, the specification describes in Paragraph [004] at pg. 4:

> . . . . For example, programs written in the high-level Java™ programming language are compiled into low level bytecode instructions. The bytecode instructions are the machine language for a Java™ Virtual Machine. The Java™ Virtual Machine Specification is described in Lindholm et al., "The Java™ Virtual Machine Specification", 1999, Addison Wesley, Second Edition.

The definitions for the instruction and operand types in the Java Programming Language based on the first letter in the opcode mnemonic are given in Linholm and so are known to those of skill in the art. Moreover, the notation is used throughout the detailed description and the type instruction and the type of operand(s) associated with the notation are described so that no more than Appellant's description is needed to establish that one of skill would know how to perform the operations recited in the claim.

The description further provides at Paragraph [007] lines 3 to 5 at pg. 5:

> . . . . In the Java™ language, type "int" is always 32 bits. Thus, 16-bit values of type "short" and 8-bit values of type "byte" are widened to the 32-bit type "int" before performing the arithmetic operation.

This description provides a direct method for comparing types and determining the relative sizes of the types, whether instruction or operand. It also explained how an instruction type is changed. Moreover, the specification provides numerous examples of type comparisons. For example,

> By way of example, if the original instruction is "iadd" which requires operands of type "int", the instruction type is set to "short" and the corresponding instruction is "sadd".

Paragraph [065], lines 5 to 7 at pg. 35.

This unambiguously establishes that the notation uses the first letter of the instruction to represent the type of instruction. Numerous other examples are provided.

> By way of example, if the instruction type is "short" and the operand type is "int", the instruction type is less than the operand type.

Paragraph [065], lines 10 to 11 at pg. 35.

> By way of example, if the instruction type is "int" and the operand type is "int", the instruction type is the same as the operand type.

Paragraph [065], lines 15 to 17 at pg. 35.

> If the instruction type is less than the original instruction type, the instruction type is set to the next

> larger type (1935). By way of example, if the instruction type is "short" and the original instruction type is "int", the instruction type is set to "int"

Paragraph [066], lines 7 to 11 at pg. 36.

> . . . For both the "sload <a>" result and "sload <b>" result operands, at 2015 the instruction type ("sadd") is not greater than the operand type ("short"), so validation of the operand types ends

Paragraph [091] at pg. 47.

> At 1905 the "idiv" instruction type ("int") is not less than the operand type ("short"). At 1915 the instruction type ("int") is also not equal to the operand type ("short") . . .

Paragraph [0105] at pg. 52.

> . . . the instruction type for "idiv" is int

Paragraph [0106] at pg. 52.

Thus, the specification unambiguously shows that instruction type and operand type are known. Moreover, the specification unambiguously teaches how to compare either an operand type or an instruction type with a specific type, such as the second type in the claims. Thus, contrary to the above quoted comments in the Examiner's Answer that there is no basis for comparing types and changing instruction types based on the comparison, one of skill in the art can determine the plain meaning of the claim language as described above.

The Examiner's Answer demonstrates that the claims are being considered in a vacuum. The MPEP directs "meaning of words used in a claim is not construed in a "lexicographic vacuum, but in the context of the specification and drawings." {MPEP §2106 at pg. 2100-7.} Had these simple directions been followed, much of the Examiner's Answer could have been eliminated. The statements concerning the level of skill and the teachings in the specification with respect to instruction type and operand type in the Examiner's Answer completely ignore both, which is an incorrect level of analysis.

The Examiner's Answer also asserts at pg. 24 that "if said operand type is less than said second type" can be ignored and an equality of types substituted for this limitation. There is absolutely no basis in the MPEP or the case law for an Examiner rewriting the claim language so as to mischaracterize an explicit claim limitation.

The claim recites, "changing the type of instructions in a chain of instructions to equal said second type." Thus, it is the type of instruction in the chain that is changed. The change is done only when a specific condition is met, "if said operand type is less than said second type."

As indicated in the above examples, if the operand type is "short" and the second type is "short," nothing would be done. If the operand type is "short" and the second type is "int," the operand type is less than the second type and so the type of the instructions in the chain is changed to "int." This interpretation follows directly from nothing other than the claim language itself in view of the knowledge of one of skill in the art as established by the specification. Thus, the comments in the Examiner's Answer about the claim language being "far fetched" at the end of Section (C) on page 21 only demonstrates that the MPEP requirements have been completely ignored and incorrect Examiner's analysis substituted for the proper claim interpretation.

When the Claims are properly interpreted, Appellant's arguments in the Appeal Brief stand unrebutted and so Claims 1 to 10, 14 to 29, 33 to 48, 52 to 67, 71 to 78 are patentable.

2. CLAIMS 11 to 13, 30 to 32, 49 to 51, 68 to 70 ARE PATENTABLE

The above comments with respect to types and comparison are incorporated herein by reference. Appellant further notes that the Examiner's Answer dismisses MPEP requirements such as using provided definitions in Claim interpretation. If the Office allows such analysis to stand, it is an admission that the MPEP has no binding effect on the examination process and as such is useless. Accordingly, Appellant respectfully submits that the parts of the Examiner's Answer that contradict the requirements specified by the MPEP should be given on weight.

In conclusion, Appellant has explained at multiple levels why the combination of references fails to render the invention as recited in Claims 1 to 78 obvious. Thus, the Examiner's rejection of Claims 1 to 78 should be reversed.

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on May 29, 2009.

Attorney for Appellant(s)	May 29, 2009 Date of Signature

Respectfully submitted,

Forrest Gunnison
Attorney for Appellant(s)
Reg. No. 32,899
Tel.: (831) 655-0880